COMPUTATIONAL METHODS FOR LARGE-SCALE MICRODATA ANONYMIZATION

A Degree Thesis
Submitted to the Faculty of the
Escola Tècnica d'Enginyeria de Telecomunicació de Barcelona
Universitat Politècnica de Catalunya
by
Xavier Casanova Colomé

In partial fulfilment
of the requirements for the degree in
Telecommunications ENGINEERING

Advisors: David Rebollo-Monedero,
Jordi Forné Muñoz

Barcelona, July 2015

RESUM

Resum — L'objectiu del control de revelació estadística de microdades és protegir la privacitat de persones i/o empreses les dades confidencials de les quals es publiquen en forma de conjunts de dades. Aquestes dades són utilitzades en estudis estadístics i, per tant, a més d'assegurar la privacitat de les persones/empreses, les dependències estadístiques de les dades publicades haurien d'ésser molt similars a les de les dades originals. La microagregació, i més concretament, la microagregació k-anònima, solventa aquest problema assegurant una preservació de la privacitat acceptable. No obstant, en termes de temps d'execució, no és convenient, ja que, encara que per conjunts de dades no molt grans el temps d'execució és acceptable, quan tractem amb conjunts de dades més grans, aquest temps s'incrementa fins a tal punt que fa el procés inviable.

Aquest informe de projecte final de grau presenta nous algoritmes que, preservant la qualitat de les dades publicades, millora el cost computacional de la microagregació k-anònima. Per a tal cosa s'utilitzen tècniques de reducció dimensional, que permeten reduir notablement el temps d'execució, i al mateix temps no comporten una gran pèrdua d'informació respecte a les dependències estadístiques de les dades que es publiquen.

Resumen — El objetivo del control de revelación estadística de microdatos es salvaguardar la privacidad de personas y/o empresas cuyos datos confidenciales se publican en forma de conjuntos de datos. Estos datos son utilizados en estudios estadísticos y, por lo tanto, además de asegurar la privacidad de las personas/empresas, las dependencias estadísticas de los datos publicados deberían ser muy similares a las de los datos originales. La microagregación, y más concretamente, la microagregación k-anonima, solventa este problema asegurando una preservación de la privacidad aceptable. Sin embargo, en términos de tiempo de ejecución, no es conveniente, ya que, aunque para conjuntos de datos no muy grandes el tiempo de ejecución es aceptable, cuando tratamos con conjuntos de datos más grandes, este tiempo incrementa hasta tal punto que hace el proceso inviable.

Este informe de proyecto final de grado presenta nuevos algoritmos que, preservando la calidad de los datos publicados, mejora el coste computacional de la microagregación k-anonima. Para tal efecto se utilizan técnicas de reducción dimensional, que permiten reducir notablemente el tiempo de ejecución, y al mismo tiempo no conllevan una gran pérdida de información respecto a las dependencias estadísticas de los datos que se publican.

REVISION HISTORY AND APPROVAL RECORD

Document distribution list	
Name	e-mail
Student: Xavier Casanova Colomé	xavier.casanova@alu-etsetb.upc.edu
David Rebollo Monedero [Project Supervisor 1]	david.rebollo.monedero@upc.edu
Jordi Forné Muñoz [Project Supervisor 2]	jforne@entel.upc.edu

Written by:		Reviewed and approved by:		
Date	25/June/2015	Date	29/June/2015	03/July/2015
Name	Xavier Casanova Colomé	Name	David Rebollo Monedero	Jordi Forné Muñoz
Position	Project Author	Position	Project supervisor 1	Project supervisor 2

ACKNOWLEDGMENT

This manuscript presents some of the results developed through the collaboration of the Universitat Politècnica de Catalunya (UPC) and Scytl Secure Electronic Voting S.A. (Scytl) in the context of the project "Data-Distortion Framework" (DDF), and in accordance with the guidelines therein. This work is thus partly supported by the Spanish Ministry of Industry, Energy and Tourism (MINETUR) through the "Acción Estratégica Economía y Sociedad Digital" (AEESD) funding plan, as a grant directly awarded to Scytl, which then served to subcontract UPC. Additional funding supporting this work has been granted to UPC by the Spanish Government through projects TEC2010-20572-C02-02 "CONSEQUENCE" and TEC2013-47665-C4-1-R "EMRISCO".

List of Figures

COMPUTATIONAL METHODS FOR LARGE-SCALE MICRODATA ANONYMIZATION

Xavier Casanova
Universitat Politècnica de Catalunya (UPC)

Abstract— Statistical disclosure control (SDC) concerns safeguarding the privacy of people and/or companies whose confidential information is released as large datasets. This data is used in statistical studies, and thus, in addition of ensuring the privacy of the individuals, the statistical dependences of the published data should be very similar to the original ones. Microaggregation, and more concretely, k-anonymous microaggregation, solves this problem with an acceptable preservation of the privacy. However, it fails in terms of execution time, since, even if for small amounts of data the required execution time is affordable, when dealing with bigger datasets the required execution time is not acceptable.

This degree project report presents new algorithms that, while preserving the quality of the released information, improve the computational cost of k-anonymous microaggregation. This is done by using dimensionality-reduction techniques, which allow to decrease noticeably the execution time, and at the same time do not incur in a significant loss of statistical dependences of the information being published.

Keywords—k-Anonymity, microaggregation, statistical disclosure control, statistical dependence, dimensionality-reduction

1. INTRODUCTION

THIS degree final project has been carried out at the Telematics Engineering department of the Escola Tècnica Superior de Telecomunicacions de Barcelona (ETSETB), at the Universitat Poitècnica de Catalunya (UPC). It has been developed within the frame of the research on security that this department performs. Some of the algorithms used had already been developed and have been provided to the author; in fact, the performed research tries to improve these algorithms.

The work here contained will be published as a journal article in the following months. The organization of this document is similar to the one used in a journal article elaboration process; however, it has been adapted to fulfill the requirements of the degree's subject TFG (*Treball de Final de Grau*, from the Catalan name of the subject).

The following paragraphs shortly present the motivation for using and developing the techniques and algorithms here explained. This section also expose the different objectives of the project, as well as the planning that was done. §2 gives an overview of the fundamentals of statistical disclosure control and includes a revision of the state of the art on k-anonymous microaggregation. Principal-component analysis and Lloyd's algorithm are reviewed in §3. In §4, the developed algorithms are presented, while §5 shows the obtained experimental results. §6 includes the budget of this project. Finally, a conclusion of our research can be found at §7.

The total amount of data in the world is expected to grow to 8 zettabyes during 2015. Moreover, it is doubling in size every two years, and, according to forecasts of International Data Corporation (IDC), it will re ach 44 zettabytes by 2020, which is nearly as many stars there are in the universe.

A good portion of this data is formed by personal information, which may be used in several ways by interested third parties, such as targeted advertising, recommendation systems, social networks, or e-voting. Additionally, all this data is used in statistical studies of all kind: a health study to find which range of age is more likely to suffer from cancer, or a study to see which are the regions where a political party has received the most votes, are just two examples of these investigations. While the utility of computerized data analysis cannot be objected, the use of confidential data poses privacy risks that cannot simply remain overlooked. On the other hand, it is precisely the availability of such sensitive data that enables the intelligent functionality these modern information technologies offer. In all of these technologies, protecting user privacy while maintaining the utility of the data necessarily supplied to possibly untrusted parties emerge as opposed objectives.

As it was shown in [11], 87% of the population in the United States had reported characteristics that likely made them unique based only on {5-digit ZIP, gender, date of birth}. About half of the U.S. population are likely to be uniquely identified by only {place, gender, date of birth}, where place is the city, town or municipality in which the person resides. In addition, even at the county level, {county, gender, date of birth} are likely to uniquely identify 18% of the U.S. population. In general, few characteristics are needed to uniquely identify a person.

The findings in [1] mean that the mere elimination of identifiers such as first and last name, or social security number, is grossly insufficient to effectively protecting the anonymity of the participants of published statistical studies containing confidential data linked to demographic information.

1.1. Objectives

As presented in the review of the state of the art in §2.2, currently several methods and algorithms to attain k-anonymity exist. All this methods accomplish the necessary condition of grouping data into groups of k records, some of them focusing in the preservation of the statistical quality of the information, and some others trying to reduce the computational time needed to run these algorithms.

Our objective is to achieve a noticeable reduction on the time needed to microaggregate a large database, and to do so we use dimensionality-reduction techniques to reduce the dimension of the dataset (here we call dimension to the number of *quasi-identifiers* in the dataset; §4.1 gives more detail about the notation used here). The use of such techniques is not without taking into account the need of preserving statistical dependence in order to use the released information in statistical studies or investigation . Indeed, quality measures are introduced to ensure that the dimensionality-reduction does not worsen too much this quality.

The main objective of our work is the introduction of principal-component analysis (PCA) in the k-anonymous microaggregation procedure, by means of three new algorithms.

More concretely, the objectives of this project are:

- Introduce dimensionality-reduction techniques in the microaggregation field, by means of principal-component analysis.
- Develop algorithms that outperform the classical approach of the problem in terms of time.

- Verify that the algorithms perform as expected, by doing several experiments.
- In the algorithms, the quality of the released information should also be taken into account.
- Make the microaggregation process more efficient in terms of memory. Dimension reduction of the data will make the algorithms more efficient in terms of memory. Indeed, after reducing the dimension of the dataset, less memory will be needed to store it. This will traduce in a more efficient use of the memory, since faster memories of the computer will be used (for example, RAM instead of hard disk).
- Develop algorithms compatible with other variations of MDAV that introduce improvements in terms of time. Indeed, the same algorithms here developed should be used with different versions of MDAV, and gains in terms of time would be multiplicative.

1.2. Planning

1.2.1. Original planning

Originally, the planned tasks were as follows.

Work Packages:

<table>
<tr><td>Project:

Computational Methods for Large-Scale Microdata Anonymization</td><td colspan="2">WP ref: 1</td></tr>
<tr><td>Major constituent:

MDAV combined with piecewise PCA</td><td colspan="2">Sheet 1 of 5</td></tr>
<tr><td rowspan="2">Short description:

Apply to the data the Lloyd algorithm and then do PCA to each one of the created cells.</td><td colspan="2">Planned start date: 06/04/2015
Planned end date: 30/04/2015</td></tr>
<tr><td colspan="2">Start event:
End event:</td></tr>
<tr><td>Internal task T1: Try the Lloyd algorithm provided by David Rebollo (the Project Supervisor)

Internal task T2: Develop my own Lloyd algorithm in Matlab

Internal task T3: Combine the use of Lloyd, PCA and MDAV algorithms

Internal task T4: Compare the performance of all the developed algorithms</td><td>Deliverables:
The Matlab code</td><td>Dates:</td></tr>
</table>

Project:	WP ref: 2

Computational Methods for Large-Scale Microdata Anonymization		
Major constituent: Alternatingly optimized PCA	Sheet 2 of 5	
Short description: Improvement of the piecewise PCA algorithm.	Planned start date: 01/05/2015 Planned end date: 25/05/2015	
	Start event: End event:	
Internal task T1: Develop the new algorithm. *Internal task T2:* test the algorithm	Deliverables*:*	Dates:

Project: Computational Methods for Large-Scale Microdata Anonymization	WP ref: 3	
Major constituent: Graphical comparison of the algorithms	Sheet 3 of 5	
Short description: Compare graphically the performance of the algorithms, in terms of time and distortion.	Planned start date: 25/05/2015 Planned end date: 01/06/2015	
	Start event: End event:	
Internal task T1: Computational time comparison *Internal task T2:* Distortion comparison *Internal task T3*: Comparison when varying the values of the MDAV cells (k), or varying the dimensions of the data, or varying the projected data energy to data energy ratio.	Deliverables*:* The Matlab code	Dates:

Project: Computational Methods for Large-Scale Microdata Anonymization	WP ref: 4	
Major constituent: Improve and optimize of the algorithms. Possibly, introduce other variations of the algorithms.	Sheet 4 of 5	
Short description:	Planned start date: 01/06/2015 Planned end date: 30/06/2015	

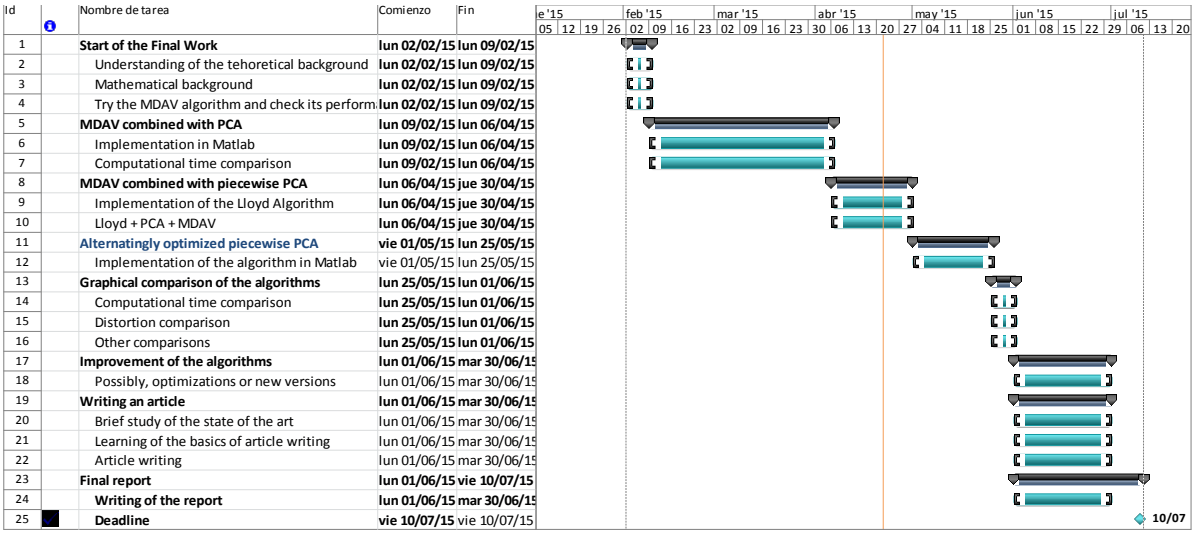
<table>
<tr><td>Improvement of the developed algorithms. These algorithms are new and don't have a predecessor, that's why we will surely have to improve them by contributing to the project with new ideas.</td><td colspan="2">Start event:
End event:</td></tr>
<tr><td>Internal task T1: Find where the algorithms spend more time.

Internal task T2: Improve the developed algorithms
*This tasks have a high degree of variability; depending on how the developed algorithms behave, we will choose what to do next. Concrete steps are not specified in this workplan since we do not know which of the ideas that we have will adjust better with the obtained results.</td><td>Deliverables:</td><td>Dates:</td></tr>
<tr><td>Project:
Computational Methods for Large-Scale Microdata Anonymization</td><td colspan="2">WP ref: 5</td></tr>
<tr><td>Major constituent:
Writing an article and the final report</td><td colspan="2">Sheet 5 of 5</td></tr>
<tr><td rowspan="2">Short description:

Journal article or conference paper.

TFG final report.</td><td colspan="2">Planned start date: 01/06/2015
Planned end date: 30/06/2015</td></tr>
<tr><td colspan="2">Start event:
End event:</td></tr>
<tr><td>Internal task T1: Writing of a version to be submitted of a journal article or a conference paper.</td><td>Deliverables:</td><td>Dates:</td></tr>
</table>

The original Gantt diagram was:

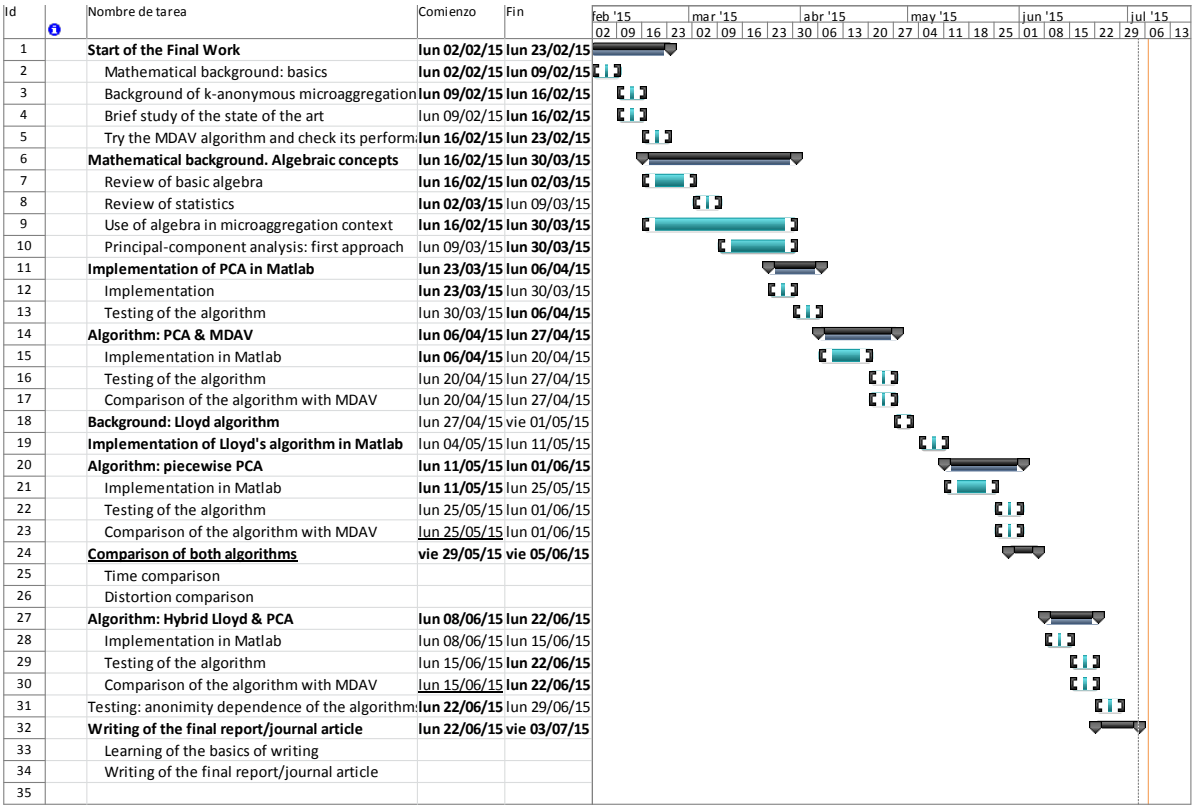


Id	Nombre de tarea	Comienzo	Fin
1	**Start of the Final Work**	**lun 02/02/15**	**lun 09/02/15**
2	Understanding of the tehoretical background	**lun 02/02/15**	**lun 09/02/15**
3	Mathematical background	**lun 02/02/15**	**lun 09/02/15**
4	Try the MDAV algorithm and check its perform	**lun 02/02/15**	**lun 09/02/15**
5	**MDAV combined with PCA**	**lun 09/02/15**	**lun 06/04/15**
6	Implementation in Matlab	**lun 09/02/15**	**lun 06/04/15**
7	Computational time comparison	**lun 09/02/15**	**lun 06/04/15**
8	**MDAV combined with piecewise PCA**	**lun 06/04/15**	**jue 30/04/15**
9	Implementation of the Lloyd Algorithm	**lun 06/04/15**	**jue 30/04/15**
10	Lloyd + PCA + MDAV	**lun 06/04/15**	**jue 30/04/15**
11	**Alternatingly optimized piecewise PCA**	**vie 01/05/15**	**lun 25/05/15**
12	Implementation of the algorithm in Matlab	vie 01/05/15	lun 25/05/15
13	**Graphical comparison of the algorithms**	**lun 25/05/15**	**lun 01/06/15**
14	Computational time comparison	**lun 25/05/15**	**lun 01/06/15**
15	Distortion comparison	**lun 25/05/15**	**lun 01/06/15**
16	Other comparisons	**lun 25/05/15**	**lun 01/06/15**
17	**Improvement of the algorithms**	**lun 01/06/15**	**mar 30/06/15**
18	Possibly, optimizations or new versions	lun 01/06/15	mar 30/06/15
19	**Writing an article**	**lun 01/06/15**	**mar 30/06/15**
20	Brief study of the state of the art	lun 01/06/15	mar 30/06/15
21	Learning of the basics of article writing	lun 01/06/15	mar 30/06/15
22	Article writing	lun 01/06/15	mar 30/06/15
23	**Final report**	**lun 01/06/15**	**vie 10/07/15**
24	**Writing of the report**	**lun 01/06/15**	**mar 30/06/15**
25	**Deadline**	**vie 10/07/15**	vie 10/07/15

Fig. 1. Original Gantt diagram

1.2.2. Updated planning

While the final version of the Gantt diagram would be:



Id	Nombre de tarea	Comienzo	Fin
1	**Start of the Final Work**	**lun 02/02/15**	**lun 23/02/15**
2	Mathematical background: basics	**lun 02/02/15**	**lun 09/02/15**
3	Background of k-anonymous microaggregation	**lun 09/02/15**	**lun 16/02/15**
4	Brief study of the state of the art	lun 09/02/15	**lun 16/02/15**
5	Try the MDAV algorithm and check its perform	**lun 16/02/15**	**lun 23/02/15**
6	**Mathematical background. Algebraic concepts**	**lun 16/02/15**	**lun 30/03/15**
7	Review of basic algebra	**lun 16/02/15**	**lun 02/03/15**
8	Review of statistics	**lun 02/03/15**	lun 09/03/15
9	Use of algebra in microaggregation context	**lun 16/02/15**	**lun 30/03/15**
10	Principal-component analysis: first approach	lun 09/03/15	**lun 30/03/15**
11	**Implementation of PCA in Matlab**	**lun 23/03/15**	**lun 06/04/15**
12	Implementation	**lun 23/03/15**	lun 30/03/15
13	Testing of the algorithm	lun 30/03/15	**lun 06/04/15**
14	**Algorithm: PCA & MDAV**	**lun 06/04/15**	**lun 27/04/15**
15	Implementation in Matlab	**lun 06/04/15**	lun 20/04/15
16	Testing of the algorithm	lun 20/04/15	lun 27/04/15
17	Comparison of the algorithm with MDAV	lun 20/04/15	lun 27/04/15
18	**Background: Lloyd algorithm**	lun 27/04/15	vie 01/05/15
19	**Implementation of Lloyd's algorithm in Matlab**	lun 04/05/15	lun 11/05/15
20	**Algorithm: piecewise PCA**	**lun 11/05/15**	**lun 01/06/15**
21	Implementation in Matlab	**lun 11/05/15**	lun 25/05/15
22	Testing of the algorithm	lun 25/05/15	lun 01/06/15
23	Comparison of the algorithm with MDAV	lun 25/05/15	lun 01/06/15
24	**Comparison of both algorithms**	**vie 29/05/15**	**vie 05/06/15**
25	Time comparison		
26	Distortion comparison		
27	**Algorithm: Hybrid Lloyd & PCA**	**lun 08/06/15**	**lun 22/06/15**
28	Implementation in Matlab	lun 08/06/15	lun 15/06/15
29	Testing of the algorithm	lun 15/06/15	**lun 22/06/15**
30	Comparison of the algorithm with MDAV	lun 15/06/15	**lun 22/06/15**
31	Testing: anonimity dependence of the algorithm	**lun 22/06/15**	lun 29/06/15
32	**Writing of the final report/journal article**	**lun 22/06/15**	**vie 03/07/15**
33	Learning of the basics of writing		
34	Writing of the final report/journal article		
35			

Fig. 2. Definitive Gantt diagram

As it can be seen, the final Gantt diagram differs slightly from the first one. The two first diagrams were a forecast of how the project would develop. However, the lack of a solid mathematical background made that I had to dedicate more time to understand these concepts, because they are necessary for understanding principal-component analysis, which is

one of the most important parts of the project. Additionally, developing the algorithms took more time than forecasted. Different errors appeared while programming them, and, even when there were no errors, the algorithms sometimes did not behave as expected. The fact of having spent more time on the two first algorithms has caused that the last algorithm has not been so fine-tuned as the other two. Anyway, the main objectives of this projects have been achieved, even if the initial predictions were a little optimistic. I have had less time than expected to write the final report, but, anyway, it has been enough.
As a conclusion on planning projects, I should be more realistic when planning the different sub-tasks, and see if any of the parts may be more difficult or problematic.

2. BACKGROUND AND STATE OF THE ART ON k-ANONYMOUS MICROAGGREGATION

2.1. Fundamentals of statistical disclosure control and microaggregation

In the statistical disclosure control (SDC) field, a *microdata set* is a database table whose records carry information concerning individuals, either people or companies. Each of these record contains attributes that may be divided into identifiers, quasi-identifiers and confidential attributes. *Identifiers* identify unequivocally the individuals. Examples of identifiers are full name or the SSNs, and they would be removed before publishing the microdata set, in order to guarantee the anonymity of the individuals. *Quasi-identifiers,* or *key attributes*, may *reidentify* the respondents if being linked with external, usually publicly available information. Examples of quasi-identifiers are age, height, weight, gender, or job. *Confidential attributes* contain sensitive information on the individuals, such as salary, political affiliation, and health condition. A simple example of microdata set is shown in Fig. 3.

Identifiers | Quasi-Identifiers | Confidential Attributes

Name	G	Age	ZIP Code	Family Income	Political Affiliation
Bob	M	14	90210	€ 57400	CiU
Betty	F	12	90210	€ 56300	PSOE
Susan	F	13	90213	€ 54100	ERC
Mary	F	18	94024	€ 39250	PP
John	M	16	94305	€ 21700	PSOE
Robert	M	17	94024	€ 32150	PP

Fig. 3. Example of dataset

As already mentioned in §1, the mere suppression of identifiers (in this case, the name) is not sufficient to ensure the preservation of privacy. Of course, identifiers are removed, but, as we will explain in the following paragraphs, the published quasi-identifiers are not exactly the same as the original ones. In Fig. 4 a diagram of the microaggregation process can be observed.

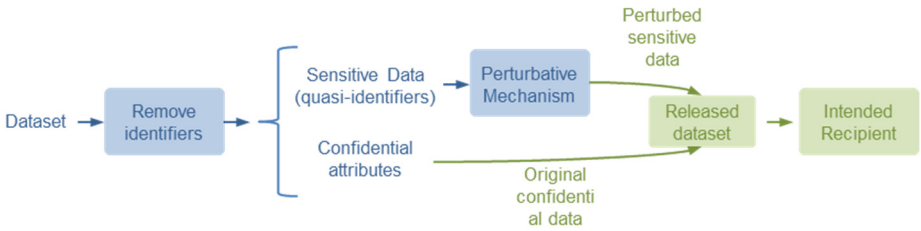


Fig. 4. Diagram of the microaggregation process

Categorical or numerical quasi-identifiers are perturbed in order to preserve privacy, at the cost of losing some of the *data utility*, seen as the accuracy with respect to the original dataset. More concretely, in k-anonymous microaggregation tuples of key-attribute values must be identically shared by at least k records. In Fig. 5, name is an identifier; gender, age and ZIP code are quasi-identifiers, and family income and political affiliation are confidential attributes. As can be seen, identifiers are removed before publishing the table. Further, the published table contains groups of k records with a common value for its quasi-identifiers. In fact, the published table is a k-anonymous version of the original one containing aggregated records. This prevents

people and/or companies from the possibility of linking unambiguously the corresponding record by combining information in the published table with information from external sources

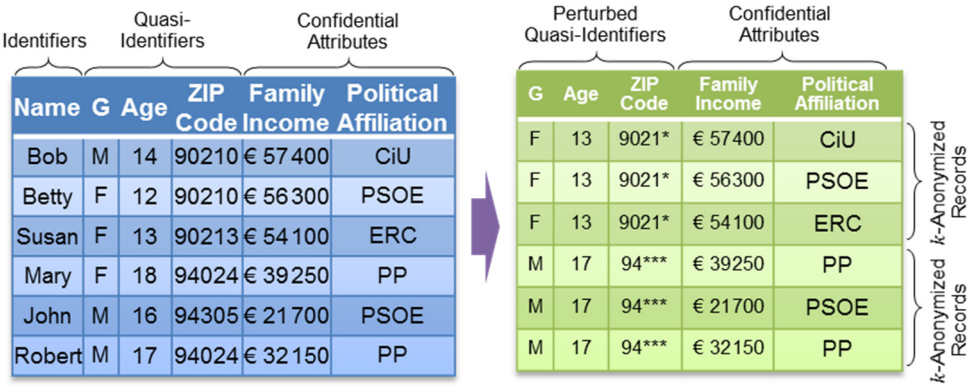


Fig. 5. Example of k-anonymous microaggregation of published data with $k = 3$.

Microaggregation algorithms are designed to perturb the quasi-identifiers in a way that the statistical quality of the published data is guaranteed. More technically speaking, microaggregation is similar to the quantization problem: the algorithms find a partition of the sequence of quasi-identifying tuples in cells of k elements, and try at the same time to reduce the distortion incurred when replacing each element in a cell by its representative within this cell. If quasi-identifiers are numerical instead of categorical, and thus, are representable as points in the Euclidean space, mean-squared error (MSE) is the usual measure of distortion.

As can be seen in Fig. 6, k-anonymous microaggregation is a similar process to vector quantization, except for the fact that it imposes the restriction that cells must be of size k (in some cases there may be a few cells with a number similar to k, depending on if the total number of records is multiple of k).

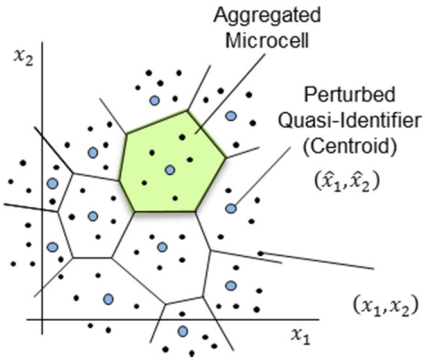


Fig. 6. Example of microaggregation, with $k = 5$. Each tuple of 5 points is assigned a representative (centroid).

2.2. State of the art on k-anonymous microaggregation

Next, we proceed to briefly review the state of the art on microaggregation, and the methods and algorithms that perform k-anonymous aggregations with reduced distortion.

Microaggregation algorithms use mean squared error (MSE) as a measure of quality (or distortion) of the microaggregated data, which can be computed as $\text{MSE} = \sum_{j=1}^{n} \|x_j - \hat{x}_j\|^2$, where n is the total number of dimensions, x is a record of the dataset (now only including quasi-identifiers), and $\hat{x}$ is the representative for this record. The microaggregation problem consists of finding a k-partition with minimum MSE.

Two kinds of algorithms exist: optimal, and heuristic. As pointed out in [2], the optimal solution to the microaggregation problem is NP-Hard, and so it cannot be solved in polynomial time;

this justifies the current use of heuristic approaches. Only in the case of univariate microaggregation the optimal can be reached, as it was shown in [3], where it is demonstrated that optimal partitions correspond to shortest paths in a graph. Even if this optimal can only be reached for the univariate case, authors of [3] claim that the polynomial algorithm can be used on multivariate data when the data vectors are projected onto a single axis.

Heuristic methods can be divided into fixed-size or variable-size of the cluster. The best-known and most widely used fixed-size algorithm is maximum distance to average vector (MDAV) [4], which is as follows:

1. Find the centroid of the dataset, find the furthest point P from the centroid, and find the furthest point Q from P.
2. Group the $k-1$ nearest points to P into a group, and then do the same with the $k-1$ nearest points to Q.
3. Repeat steps 1 and 2 on the remaining points until there are less than $2k$ points.
4. If there are k to $2k-1$ points left, form a group with those and finish. Else, if there are 1 to $k-1$ points, adjoin them to the last (hopefully nearest) group.

MDAV has proven to be a good performer in terms of time (at least for not very large databases) and one of the best regarding the homogeneity of the resulting groups. In [5] V-MDAV, a variation of MDAV, is presented. It is a variable-size heuristic algorithm, and overcomes the fixed group size constraint of MDAV with a similar computational cost.

Sorting by the first principal component is another method to microaggregate in a faster manner, as described in [6] and in [7]. In this case, data is projected onto a single axis, the principal component, that is, the one containing most of the information. In cases where one dimension keeps almost all the information, this procedure may achieve good performance. However, in a general case, where there is no guarantee that this happens [6], a different strategy should be adopted. In our work, we propose the use of more general dimensionality-reduction techniques. Namely, we propose the use of multidimensional principal-component analysis (PCA) instead of the one-dimensional PCA that had already been used in the past.

The computational cost of MDAV is quadratic with the number of records of the dataset. More concretely, it can be shown that the computational cost of MDAV can be expressed as $\frac{n^2}{k}(m+m_0)$, with $m_0 \simeq 4$ and dependent on the computer on which the algorithm is run.
As explained, when the number of records is too big, the microaggregation process becomes unfeasible. Additionally, when the total number of dimensions m decreases, the computational cost also becomes smaller. That is why, as explained in the next section, we introduce dimensional reduction techniques in the microaggregation process.

3. PROPOSED ALGORITHMS TO SPEED-UP MICRODATA ANONYMIZATION

Two well-known algorithms have been used in this work: principal-component analysis[10] and Lloyd's algorithm [9]. This section includes a brief review of both algorithms.

3.1. Principal-component analysis

The main innovation in our algorithms is the introduction of dimensionality reduction techniques in order to reduce the computational time required to carry out the microaggregation process. This is done by means of principal-component analysis (PCA), a widely used technique in machine learning. By using PCA we obtain the representation of the points in the original vector space into the vector subspace of reduced dimension. Indeed, this representation is the orthogonal projection of the dataset onto the vector subspace.

In order to make the explanation easier, let us consider that the dataset is made up of n records (n points) of m dimensions each one (m quasi-identifiers), and that all of them are contained in a matrix X, as Fig. 7. Matrix representation of the dataset. shows.

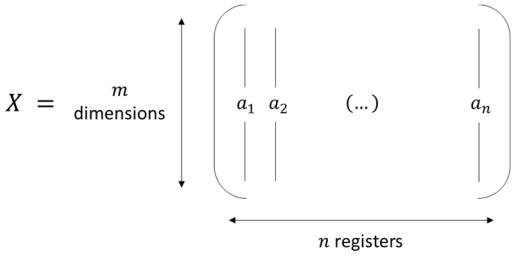


Fig. 7. Matrix representation of the dataset.

where a_i , $1 \leq i \leq n$ is an m-dimensional record. Fig. 8. Block diagram of PCA presents a block diagram of how the algorithm works. Firstly, matrix $\widetilde{\mathrm{U}}^{\mathrm{T}}$ is applied to the dataset X. This results in a compressed version of X, which is actually its projection onto the vector subspace of dimension m' (the criteria followed to determine m' is explained in the following paragraphs). This dimension-reduced version of X is called $\tilde{X}$, and it equals $\widetilde{\mathrm{U}}^{\mathrm{T}} \cdot (X - \mu_x)$, where μ_x is the mean of X.

$\tilde{X}$ can be 'reconstructed' to the original m-dimensional vector space as follows: $\hat{X} = (\tilde{U} \cdot \tilde{X}) + \mu_x$. This 'reconstructed version' of X is actually the projection of X onto the vector subspace of dimension m.

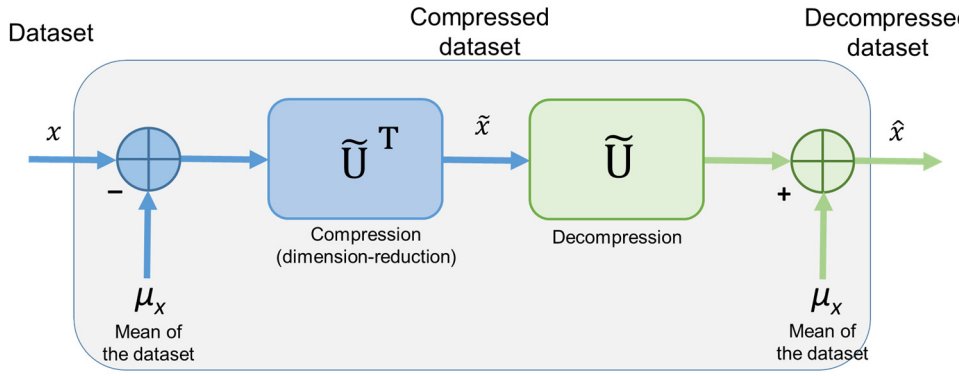


Fig. 8. Block diagram of PCA

Matrix $\widetilde{\mathrm{U}}$ can be obtained by following this procedure:

1. Subtract the mean.
2. Compute the covariance matrix $\Sigma = (X \cdot X^T)/n$.

3. Compute the spectral decomposition of the covariance matrix, $\Sigma = \mathrm{U} \cdot \Lambda \cdot \mathrm{U}^{\mathrm{T}}$ (note the notation used here: matrix U is not the same as matrix $\widetilde{\mathrm{U}}$), where Λ is a diagonal matrix containing the eigenvalues of Σ and U contains the corresponding eigenvectors. In Fig. 9, v_i, $1 \leq i \leq m$, are eigenvectors of Σ, and $\lambda_i, 1 \leq i \leq m$ are its corresponding eigenvalues.

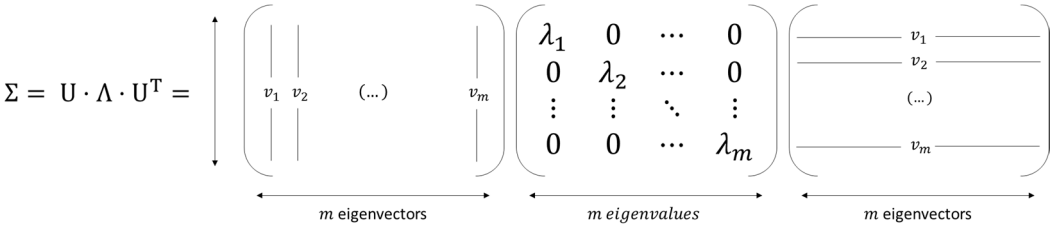


Fig. 9. Spectral decomposition of the covariance matrix.

4. Sort the U matrix into descending order of the eigenvalues contained in Λ.

Before continuing the explanation, let us define the energy ratio after applying the matrix $\widetilde{\mathrm{U}}^{\mathrm{T}}$ to the dataset as

$$E = \frac{\lambda_1 + \cdots + \lambda_{m'}}{\lambda_1 + \cdots + \lambda_m},$$

where $m' \leq m$ and $0 \leq E \leq 1$.

5. Given an energy ratio, calculate the compression matrix $\widetilde{\mathrm{U}}$ as the first m' columns of U.

For the purpose of this work, PCA was implemented in Matlab. In Fig. 10 PCA has been applied to a synthetically generated dataset containing 500 2-dimensional points (`randn(2,500)`).

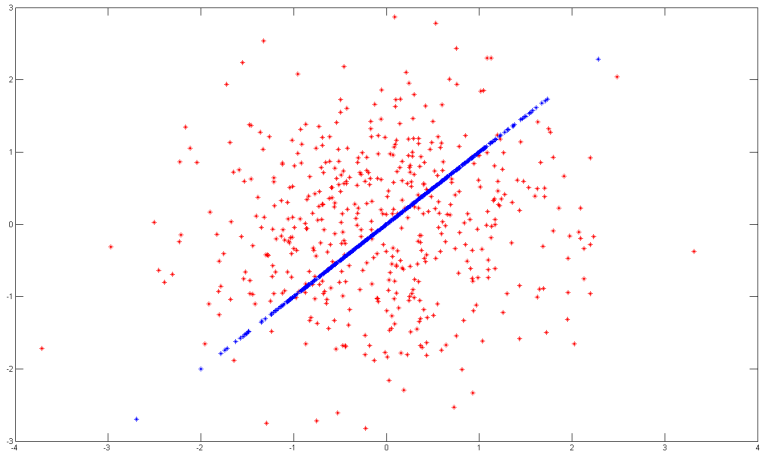


Fig. 10. Use of PCA in a 2-dimensional dataset.

Points in the original 2-dimensional vector space are in red, while projected points in the 1-dimensional space are in blue. The blue line is in fact the orthogonal projection of the original points, and, in spite of having used an energy ratio of 0.45, the result is still accurate.
The same can be observed in Fig. 11 and for a 3-dimensional dataset.

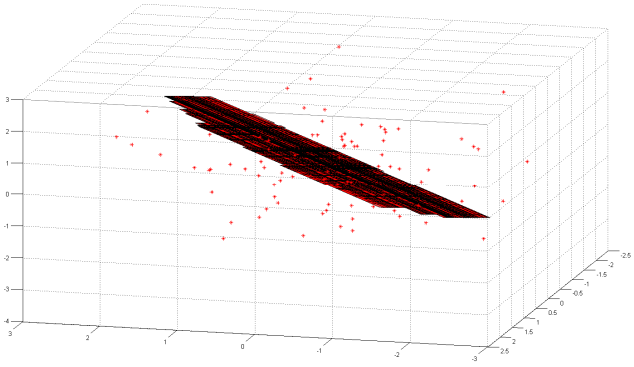


Fig. 11. Use of PCA in a 3-dimensional dataset.

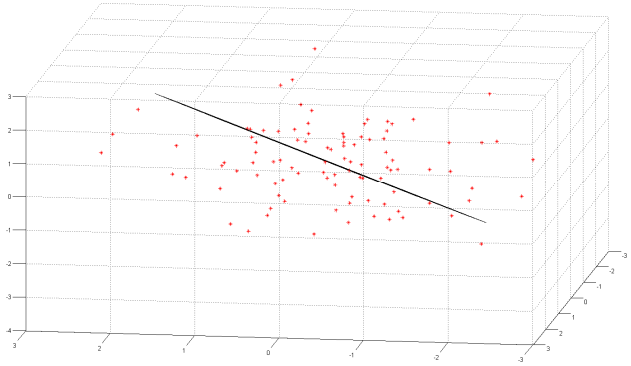


Fig. 12. The same projection as in Fig. 11 seen from another point of view.

Now 3-dimensional points are projected onto a 2-dimensional vector space, which is, indeed, a plane.

We have already introduced principal-component analysis. This algorithm will be used in order to reduce the total number of dimensions of the dataset, and, as it has been explained, an important parameter for the algorithm to work properly is the energy ratio. In fact, for a given energy ratio, the number of dimensions of the projected version of the dataset will depend on the concrete dataset. Indeed, the energy ratio depends on the eigenvalues that are obtained for each dataset, and thus, different datasets could result in different total number of dimensions of its projected version, even if the energy ratio were the same.

Let us illustrate this with an example. Fig. 13 shows the eigenvalues for the dataset Census. For each dimension, the energy on this dimension and the cumulated energy are shown. As it can be observed, with only 1 dimension we keep 58,7% of the energy, while only with 7 dimensions we would keep 98,3% of the energy. Thus, with 7 dimensions we would obtain a good representation of the dataset, or, what is the same, a good preservation of the quality of the information contained in it.

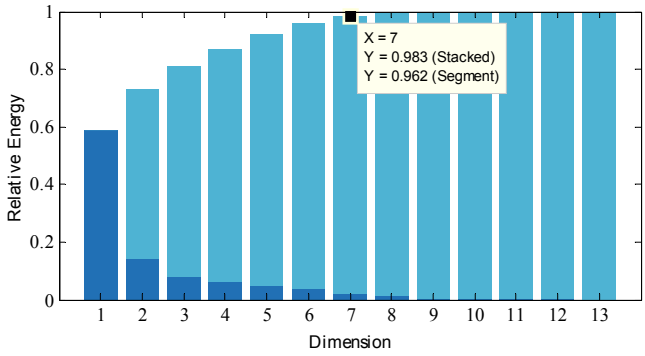


Fig. 13. Energy per dimension and cumulated energy.

3.2. Lloyd's algorithm

Lloyd's algorithm [9] is used in two of the three developed algorithms. This section describes it briefly and presents some graphical results of the generated Matlab code.

Lloyd's algorithm is a well-known and widely used algorithm. It is used to partition a set of points into cells and assign a representative to each point within a cells, in an iterative way, and ensuring that the resulting MSE (distortion) is minimal. This minimal value can be a local value or an absolute value, and some tests to the algorithm should be done in order to see its behavior regarding the resulting distortion.

Lloyd's algorithm follows these steps:

1. Choose a number of centroids equal to a given number of cells.
2. For each point, find which is the nearest centroid (in terms of MSE), and assign it as a new representative for the point (nearest-neighbor condition).
3. For each cell, recalculate the centroid (centroid condition) as the mean of all the points within the cell (a cell is made up of all the points with the same centroid).
4. Repeat steps 2 and 3 until the relative change on the distortion is under a certain given threshold.

To avoid that the process lasts too much time, a maximum number of iterations is defined in the algorithm. This number is defined after trying different values for each application and observing that results are coherent (in our case, typical values are between 15 and 25). Additionally, if after an iteration the cells remain the same, the looping also finishes, because this means that distortion will not change any more.

For the purpose of this work, Lloyd's algorithm was implemented in Matlab. The following example shows the resulting cells for 1,000 2-dimensional points.

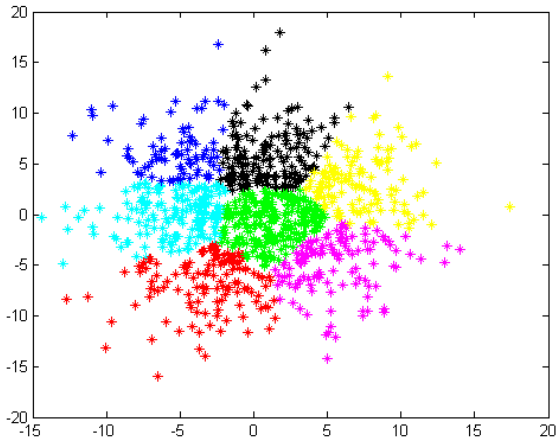


Fig. 14. Example of application of Lloyd's algorithm.

Each point have a different color in function of the cell where it is. Cells are called Voronoi cells.

4. PROJECT DEVELOPMENT

This section presents the three k-anonymous microaggregation algorithms that have been developed, as well as the contribution of each one. §4.1describes the notation used in this work.

4.1. Notation

Recalling the already explained notation, in our work, X is the dataset to be microaggregated, and it is expressed as a matrix as shown in Fig. 15.

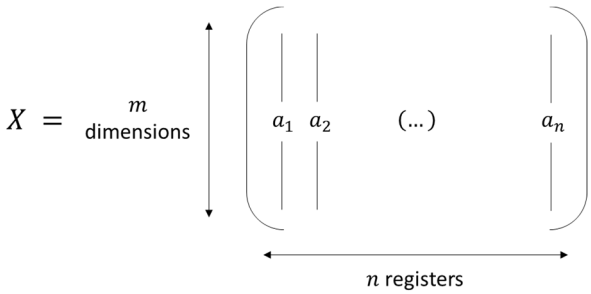


Fig. 15. Dataset expressed as a matrix

Under this notation, a_i , $1 \leq i \leq n$ is an m-dimensional record. Note that the total number of dimensions m, is the same as the total number of *quasi-identifiers*, which should be perturbed in order to preserve privacy of the individuals. In addition, n is the total number of records, that is, the total number of individuals in the dataset.

The reduced-dimension version of the dataset is denoted by $\tilde{X}$, and it is also expressed as a matrix in Fig. 16.

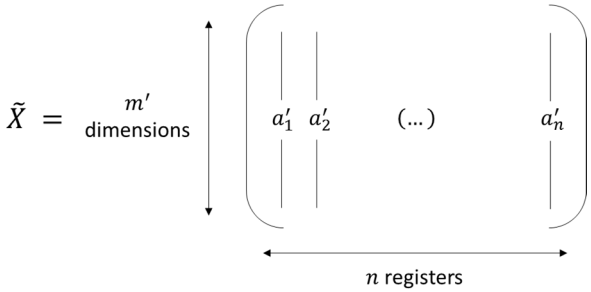


Fig. 16. Reduced-dimension version of the dataset

4.2. Dimensional reduction of the data

Dimensionality reduction is a widely used technique in machine learning, and that is why the idea of reducing the total number of dimensions of the dataset came as natural while trying to find new methods to microaggregate in a faster manner. Dimensionality reduction is often performed by means of principal component analysis (PCA). Different algorithms involving the use of PCA and MDAV have been developed; all of them are presented in the next sections.

PCA projects data from the original m-dimensional vector space onto a reduced-dimension vector space. The dimension of the latter depends on each particular dataset, and it is chosen under a criteria ensuring a small information loss (small distortion). PCA projects data by doing the product between the dataset and a projection matrix; the projection matrix can be obtained from the decomposition of the covariance matrix of the dataset, $\Sigma = \mathrm{U} \cdot \Lambda \cdot \mathrm{U}^{\mathrm{T}}$, where Σ is the covariance matrix of the dataset, Λ is a diagonal matrix containing its eigenvalues and

U contains the corresponding eigenvectors (the reader can find more information on how PCA works in §3.1).

It must be noted that the use of this technique has more impact in cases where the dataset has a great part of the information stored in few dimensions (e.g. in 4-6 dimensions). Fig. 17 shows the eigenvalues in each dimension (energy per dimension) and the cumulated energy.

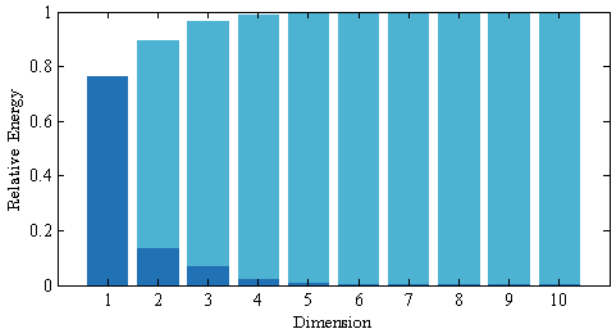


Fig. 17. Normalized energy for the 'EIA' dataset. 99,5% of the energy is kept with only taking 5 dimensions.

4.3. PCA & MDAV

The first algorithm consists of two main steps:

1. Perform a principal component analysis to the dataset (obtaining $\tilde{X}$)
2. Execute MDAV on the dataset of reduced dimension $\tilde{X}$.

In step 1, matrix $\widetilde{\mathrm{U}}^{\mathrm{T}}$ is applied to the dataset X. This results in $\tilde{X}$, which is actually the projection of X onto the dimensionally-reduced vector subspace (the reader can find more information on how PCA works in §3.1).

As the reader may have already noticed, now MDAV is executed on the reduced-dimension vector space. The new anonymized data will not be represented with the same precision as when only MDAV was used, because MDAV will calculate the clusters from the projected version of the dataset. However, as the projected dataset is very close to the original one in terms of quadratic distance (MSE), the distortion incurred in this case should not be very different, even if the difference with distortion from only MDAV may still be observed. This loose in distortion is justified by a decrease of the amount of time needed to execute MDAV, since now we are operating in a dimension-reduced vector space, which makes the operations significantly faster. For example, a dataset with 10 dimensions may be reduced to a 5-dimensional vector space with an energy ratio of 99,5%, as shown in Fig. 17.

4.4. Piecewise PCA

The second algorithm, which we have named *Piecewise PCA*, does something similar to what *MDAV & PCA* does. This algorithm, however, should require still less time than *MDAV & PCA* while incurring in a lesser distortion than the latter. It combines the use of the Lloyd's algorithm with the principal-component analysis.

The main steps that the algorithm follows are:

1. Run Lloyd's algorithm on the dataset, and thus divide it into a determined number of cells.
2. Perform PCA on each cell.
3. Execute MDAV on each cell.

4. In this case, total distortion will be calculated as the sum of the distortions in each cell, weighted by the number of points in the cluster with respect to the total number of points in the dataset.

This way of partitioning the dataset should give in this case a better representation of the dataset in the dimension-reduced vector space. Note that now each cluster will have a different reduced-dimension space where data will be represented. Is in each one of these clusters, after performing PCA, that the data is anonymized with MDAV. The new algorithm should reduce considerably the execution time, because of the MDAV's execution time quadratic dependence with the number of registers; indeed, now the number of points in each cluster is fewer than the total number of points, and this give, in addition of a better distortion thanks to the better representation of the data, a smaller execution time.

4.5. Hybrid Lloyd & PCA

The third and last algorithm combines the use of Lloyd's algorithm with principal-component analysis. The main goal of this algorithm is to obtain a final distortion smaller than the one obtained with the two precedent algorithms.

Basically, this algorithm adapts the partitioning of the dataset into clusters that will be represented in a more accurate manner in terms of PCA. The main steps of the algorithm are as follows:

1. Apply Lloyd's algorithm to the dataset (to make the explanation easier, consider that 2 clusters are formed).
2. Perform a principal-component analysis into each Lloyd's cluster. Now each point of the dataset is represented into a reduced vector space. It could happen that a point that per Lloyd's algorithm is in one cluster, was better represented in terms of PCA if it was in the other cluster. Here is what this algorithm takes advantage of.
3. For each point, perform PCA to project them into each one of the two vector subspaces (one for each cluster).
4. Compute the distances (MSE) of each point to its representation into both vector subspaces.
5. Choose as a new cluster for each point the one where its PCA representation gives the smallest distortion (smallest MSE).
6. Repeat steps 2 to 5 until the distortion relative change reaches a given minimum.
7. When step 6 is finished, two (or n in a more general case) clusters exist. As was done in the algorithm described in §4.4, *Piecewise PCA*, perform a principal component analysis in each cluster.
8. Execute MDAV into each cluster.
9. Calculate the distortion as the sum of weighted distortion (same as in *Piecewise PCA*).

This algorithm adapts the dataset into clusters that are optimal in terms of PCA representation. This means that when PCA is finally applied, the representation gives the smallest distortion incurred by the use of PCA among all the algorithms here described. Thus, in addition of reducing the execution time by reducing the dimension of the dataset, the incurred distortion is not a high price to pay, because the data is well adapted to its new representation.

5. EXPERIMENTAL RESULTS

After having explained the three algorithms and the techniques used to achieve a reduction in the execution time, here we present some experimental results obtained after running them. Different datasets have been used, with different number of records (points) or varying the dimension (number of quasi-identifiers) of the reduced-dimension vector subspace.

5.1. Results for PCA & MDAV

The first algorithm is directly compared against the case in which only MDAV is used. Fig. 18 shows the dependence of the execution time with the number of records of the dataset.

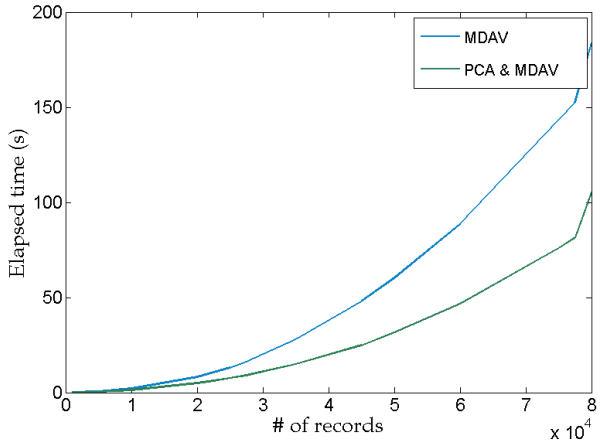


Fig. 18. As the number of records increases, the difference between classical MDAV and PCA & MDAV becomes greater.

In the graph in Fig. 18, MDAV and PCA & MDAV have been run with different number of records, all of them taking samples from the dataset 'Large Census' (`datasample(XDatAtXDimAtX,75000,2,'Replace',false))`. As it can be observed, the new algorithm clearly outperforms MDAV in terms of time, and the gain becomes greater as the total number of records increases.

Decreasing the running time is important, but an algorithm performing better in terms of time should also not worsen too much the quality of the information, or, what is the same, should not increase the distortion too much. In Fig. 19 the obtained distortion for the same cases as in Fig. 18 can be observed.

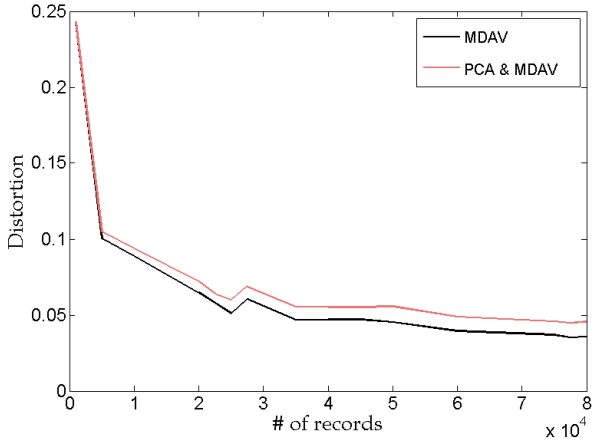


Fig. 19. Dependence of the distortion with the total number of records in the dataset.

It must be remarked that, as a convention, the total variance of the dataset is normalized, so that the variance in each dimension is 1. That is the reason why distortion takes these values.

The fact that distortion decreases with the total number of records is coherent. Indeed, if more points (records) are available, it is more likely that cells contain points that are nearer that in a case with less points. Therefore, as cells contains points that are close to each other, the centroid will also be near to this points, and so, the total distortion will be smaller as the number of records increases.

Also, in the example, note that when the number of records is 27,500, the distortion increases a little bit. This experiment has been done from a unique dataset, 'Large Census', containing 149,642 13-dimensional records, and by taking samples from it. This does not ensure that the choice of points done by Matlab in a random manner is the best for this purpose; it may happen that points are not close enough, and thus the distortion may be slightly greater. However, as it can be seen, distortion tends to decrease.

In this experiment the algorithm behaves as expected. This is due to the fact that most of the information is in the first 5 dimensions of the dataset, which allows the algorithm to take great advantage of the dimensionality reduction (from 13 to 5 dimensions). In Fig. 20 a case where more dimensions are needed to retain the same information is shown. Fig. 21 shows the distortion for the same dataset (in this case, the dataset is 'Quant Forest').

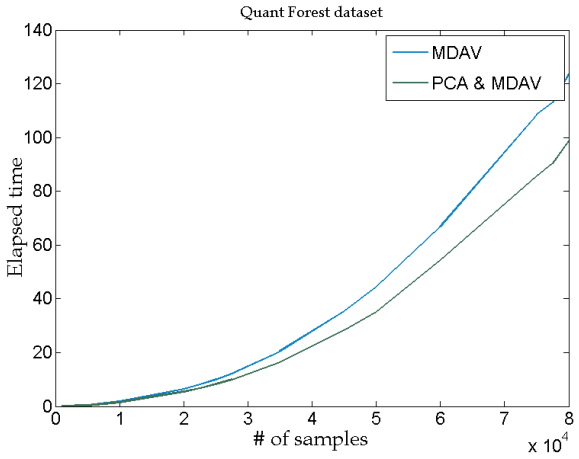


Fig. 20. There is still a noticeable difference between both algorithms, but this time it is smaller.

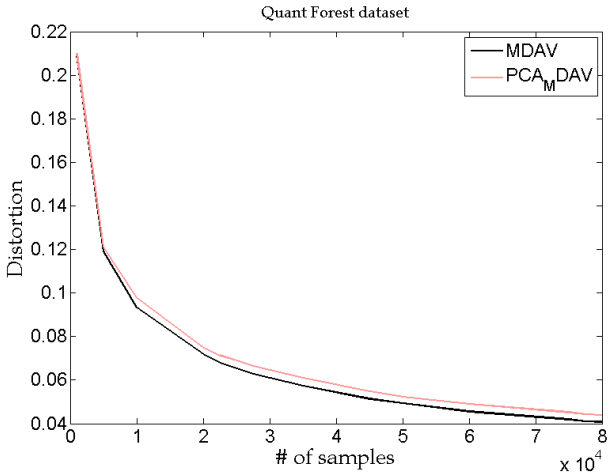


Fig. 21. Again, distortion is not much greater in the case of PCA & MDAV.

In order to understand why in this case the gain in terms of time is smaller, we include in Fig. 22 the relative energy per dimension of the 'Quant Forest' dataset.

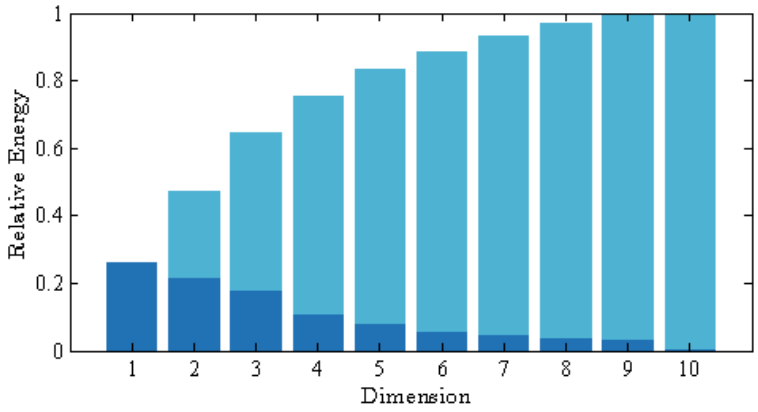


Fig. 22. Energy per dimension and cumulated energy.

In this case (Quant Forest dataset) more dimensions are needed to obtain the same quality as in the 'Large Census' dataset case. We can still observe here that the new algorithm, PCA & MDAV, has a better performance in terms of time, but this time the difference with MDAV is smaller. This happens because now the number of dimensions needed to attain the same quality is greater, and so, the algorithm cannot take as much advantage of the dimensionality reduction of the dataset. Again, distortion is slightly greater that when only using MDAV, but it is almost the same.

The performance of this algorithm also depends on the energy ratio used in PCA (the reader can find more information about PCA in §3.1), since it determines the number of dimensions of the projected dataset. This can be observed in Fig. 23.

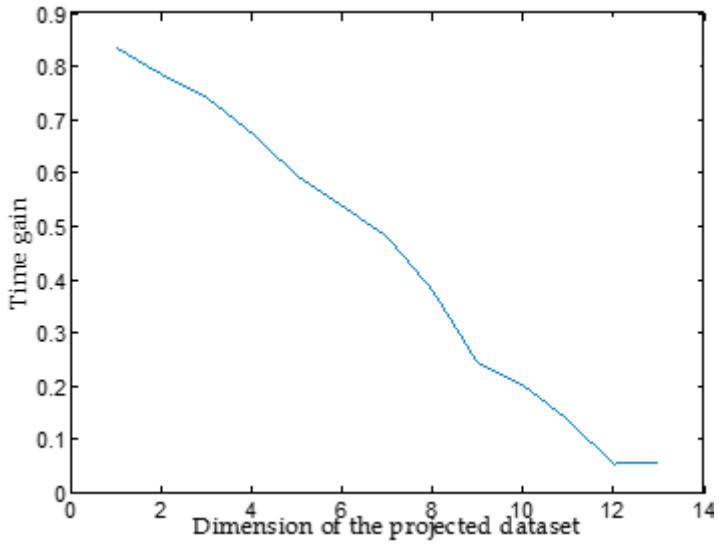


Fig. 23. Time gain of PCA & MDAV in front of MDAV.

In Fig. 23, the gain in terms of time of this algorithm is presented. When the number of dimensions decreases, the gain becomes higher, because MDAV spends less time in its execution. When the number of dimensions is 13 (as the original dataset), the time gain is almost 0. It is not exactly 0 because the reconstructed version of the dataset is not exactly the same as the original one.

The same comparison is done in terms of distortion in Fig. 24.

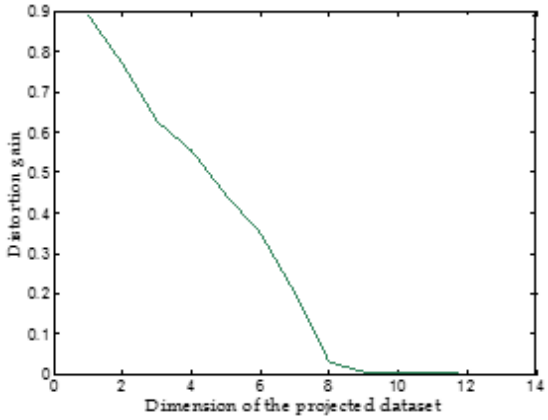


Fig. 24. Distortion loss gain of PCA & MDAV in front of MDAV.

As it can be observed in Fig. 24, if we keep 8 dimensions, the distortion incurred with this algorithm is almost the same as with MDAV. Even if we took 7 dimensions, the difference will be small. Furthermore, this is only the distortion gain; the absolute values were already small, but here the difference between both algorithm is shown. Note the notation used here: distortion gain would be in this case a loss on distortion, because in fact the algorithm's distortion is slightly worse than the MDAV's one.

This algorithm clearly outperforms MDAV. As it has been exposed, execution time is certainly smaller when using PCA before MDAV, and the distortion incurred, even if slightly greater, is almost the same. The performance of the algorithm will rely ultimately on the amount of information per dimension of each dataset. In consequence, we can affirm that this is a good algorithm to be used in cases where the dataset has these desirable properties.

5.2. Results for Piecewise PCA

In this section we compare the second developed algorithm with the original MDAV. The performance in terms of time for the dataset 'Large Census' can be observed in Fig. 25

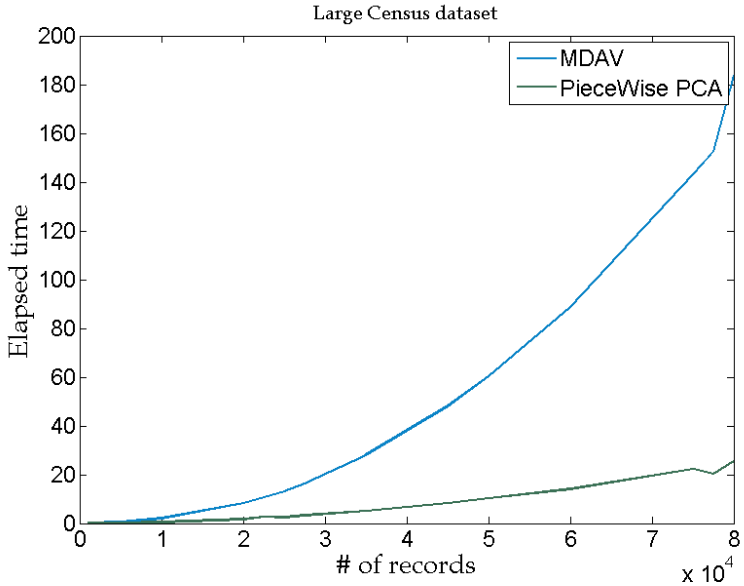


Fig. 25. Piecewise PCA spends still less time than PCA & MDAV.

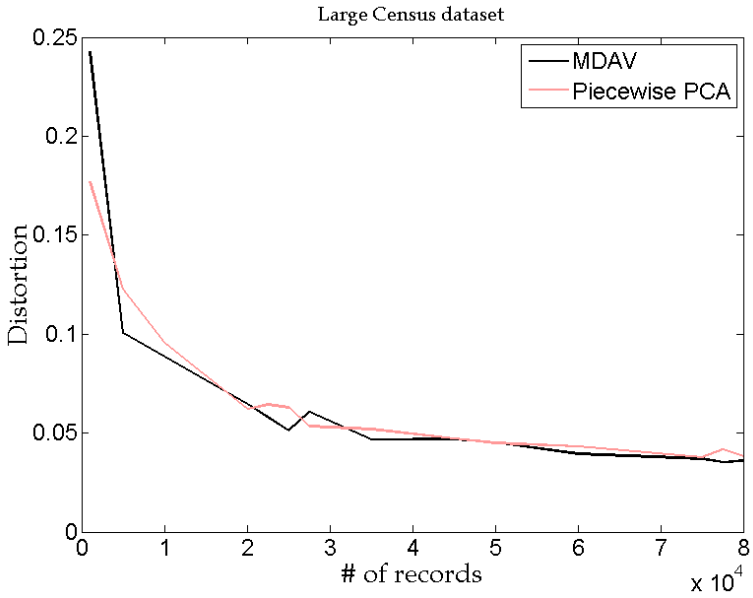


Fig. 26. Piecewise PCA: distortion in function of the total number of records, for the case of 'Large Census' dataset.

There is no loss in distortion in Piecewise PCA, as can be seen in Fig. 26 (and even distortion may be sometimes a little better), and so, this algorithm preserves the quality of the information that is to be released.

The performance of this algorithm is not compared against the dimension of the vector subspace since it takes different values for each Lloyd's cluster.

Yet, here the partitioning of the data with the Lloyd's algorithm "hides" the contribution of PCA in the execution time reduction. That is why we will also compare Piecewise PCA algorithm with two similar algorithms which do the same but without PCA:

- Algorithm A (Fig. 27Fig. 28) partitions the data with Lloyd's algorithm and then microaggregates each cell with MDAV.
- Algorithm B (Fig. 28) consists in partitioning the dataset with MDAV instead of with Lloyd, and then microaggregating each cell with MDAV.

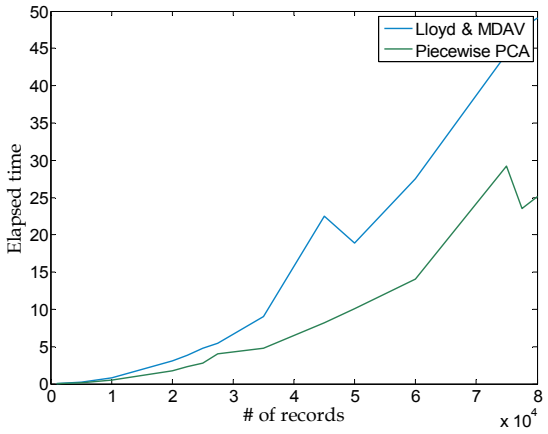


Fig. 27. Contribution of PCA to the time gain in Piecewise PCA.

Observe that when only Lloyd is used, the algorithm is still faster than MDAV. However, if we add the use of PCA (and thus, we use Piecewise PCA), it is still faster, as shown in Fig. 27.

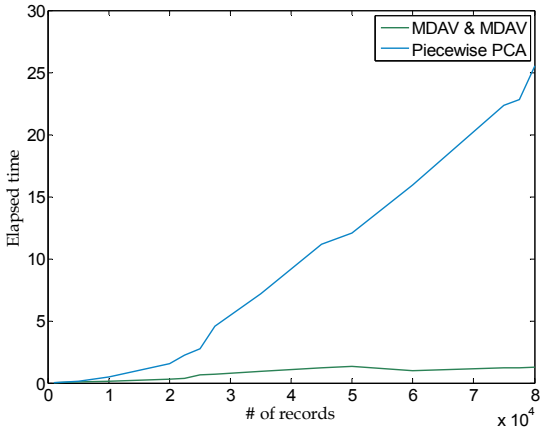


Fig. 28. Contribution of PCA to the time gain.

In Fig. 28, algorithm B has been compared against Piecewise PCA. Again, the use of PCA achieves better execution times.

In terms of time, this algorithm clearly outperforms MDAV, and also PCA & MDAV. When the number of records is small, there is not a great difference between Piecewise PCA and MDAV. However, the effect of partitioning the data into cells with Lloyd algorithm makes that, when the number of records is bigger, there is a clear difference between both algorithms. In fact, with Lloyd algorithm we are reducing considerably the number of records where microaggregation must be performed, and so, the dependence with the number of records has less impact.

The performance of this algorithm is still better than PCA & MDAV, since it achieves smaller execution times with a total distortion similar to the MDAV case.

5.3. Results for Hybrid Lloyd & PCA

Due to time restrictions, this algorithm is not totally finished. Some work is still to be done in order to fine tune its behavior and performance. Yet, we also present in Fig. 29 and Fig. 30 the provisional obtained results .

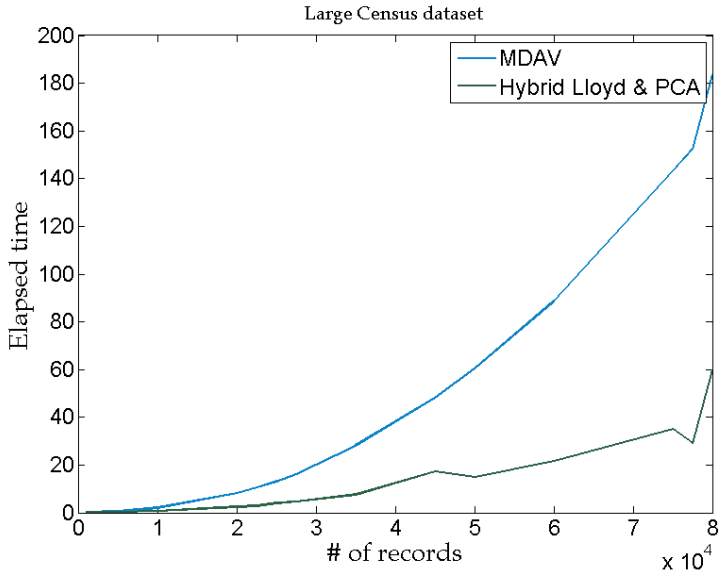


Fig. 29. Dependence of the execution time with the total number of records.

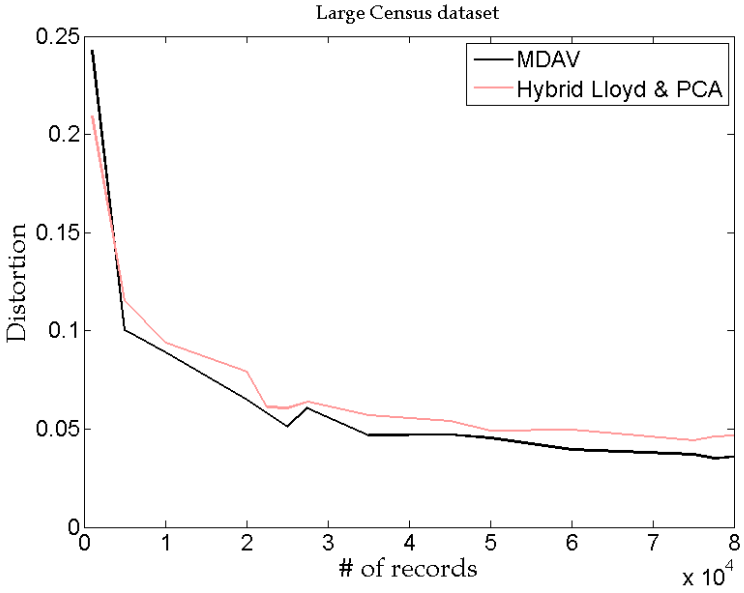


Fig. 30. Dependence of distortion with total number of records.

Here the execution time is also smaller, because of the pre-partitioning of the data. In this case, distortion is also similar to the MDAV case; however, this algorithm was conceived to attain a smaller distortion by adapting the partitioning to the PCA representation of the data. That is why the algorithm is not finished and should still be fine-tuned in order to achieve this better partitioning of the data. This is part of the future work to be done in the framework of this project.

5.4. Anonymity dependence: comparison between PCA & MDAV and Piecewise PCA

To finish with the results presentation, we compare here the performance of both PCA & MDAV and Piecewise PCA in function of the anonymity desired (the number of elements within each MDAV cell, the k in the k-anonymous microaggregation).

Firstly, Fig. 31 shows the anonymity-dependence of the execution time.

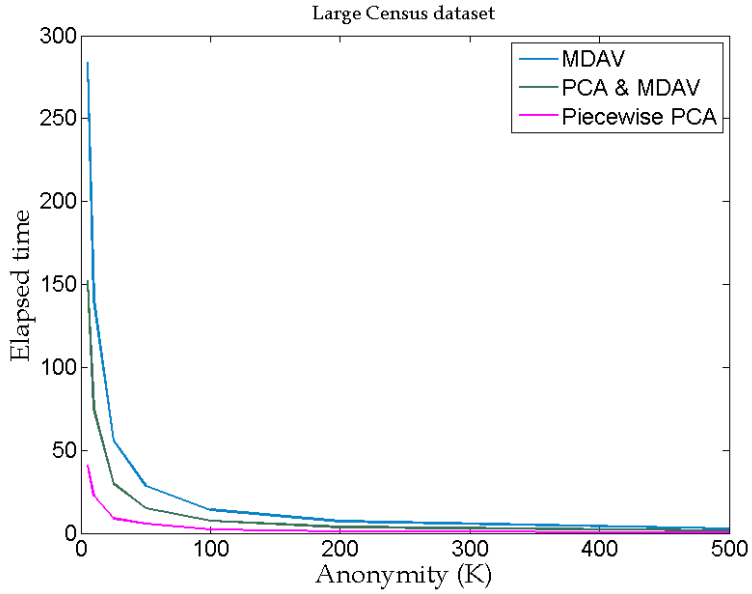


Fig. 31. The new algorithms have a smaller execution time than MDAV.

As it has been already explained in §5.1 and §5.2, Piecewise PCA is the best algorithm in terms of time, and PCA & MDAV is better than MDAV. It can be observed that when the anonymity increases, the execution time decreases. A bigger k means that the cells in MDAV have more points, and so, the total number of cells in MDAV is smaller. This means that MDAV has to spend less time in creating these cells. Since MDAV is the bottleneck in k-anonymous microaggregation, the execution time decreases noticeably with higher values of k.

In Fig. 32 distortions are similar in the three cases, and Piecewise PCA usually has a lesser distortion than the other two.

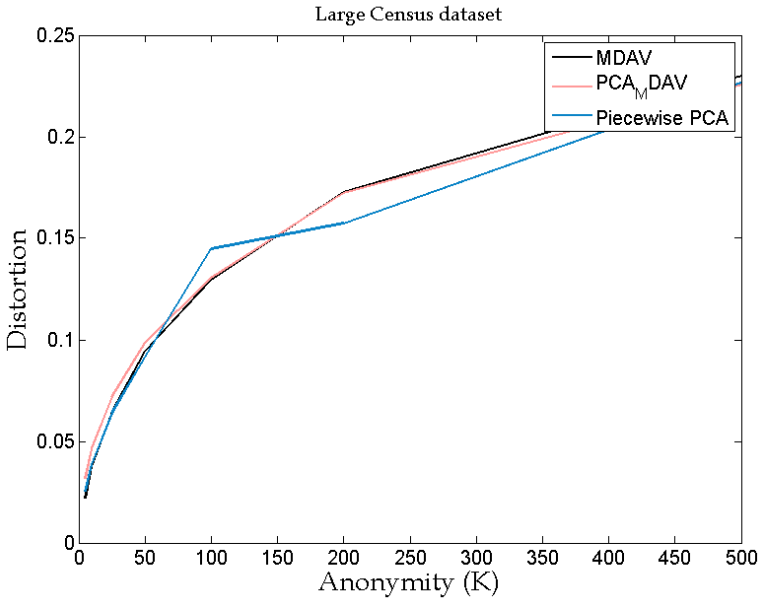


Fig. 32. Anonymity-dependence of the distortion.

As expected, distortion increases with k in the three cases. To understand why this happens, consider a dataset with 1,000 records and $k = 1{,}000$. This would mean that there is only one MDAV cell, and that all the records of the dataset would be represented by a single centroid (a single representative). This would give a great distortion, since a great part of the records would be far away from this centroid, and so, the MSE would be high. Now consider a dataset with 1,000 records and $k = 1$. In this case, there would be as many MDAV cells as records in the dataset. Thus, the centroid for each record would be the record itself, giving a distortion equal to zero. This behavior is well represented in the graph in Fig. 32.

6. BUDGET

In this project, no hardware has been needed. Only new algorithms have been developed, and thus, the total economic cost of the project should be measured in terms of the hours spent by each one of the project members. Below is the breakdown of hours spent by each one, as well as the total cost of the project.

Project member	Number of hours	Hourly cost (€/hour)	Total cost
Xavier Casanova (project author)	468	15.00*	7020.00 €
David Rebollo Monedero (project supervisor 1)	156	34.56	5,391.36 €
Jordi Forné Muñoz (project supervisor 2)	78	37.66	2,937.48 €

In conclusion, the total cost of this project has been 15,348.84 €.

*This value is an estimation.

7. CONCLUSIONS AND FUTURE DEVELOPMENT

Protection of the anonymity of people and/or companies whose confidential data is released is a crucial problem nowadays. A great volume of this information is used every day in statistical studies of all kind, and thus, having methods to ensure this anonymity is of paramount importance.

A widely used technique to protect the anonymity is k-anonymous microaggregation. Its traditional approach is simply using the algorithm maximum distance to average vector (MDAV). MDAV is an heuristic algorithm which attains k-anonymity with a good preservation of the statistical quality of the information. However, even if with small datasets the microaggregation process is feasible in terms of time, when dealing with larger datasets, the execution time is not acceptable.

In this work, dimensionality-reduction has been introduced in the microaggregation field by means of principal-component analysis. All the proposed algorithms include the use of this technique, and have shown to spend less time than MDAV to microaggregate large datasets. This gain in the execution time comes with a distortion that is similar to the classical approach, and thus, this is an important achievement in the statistical disclosure control field. Differently from the approaches that had been done in the past, the dimension of the projected version of the dataset has been taken into account, in order to preserve the quality of the information to be released. Indeed, the distortion loss is negligible in front of the time gain if enough dimensions are kept.

The third algorithm does not show the expected behavior. In this case, the objective was to improve the incurred distortion, but, however, distortion is slightly higher than in the two other cases. Execution time is smaller, but this was not the main objective of this algorithm. Fine-tuning this algorithm is part of the future work to be done within the framework of this project.

All the algorithms here presented use the classical approach of MDAV as one of the steps of the process. Thus, different versions of MDAV could be used inside these algorithms, and they would remain the same. This will allow, in future versions of the algorithms, to include the use of improved versions of MDAV in our algorithms, with no need of modifying them.

In addition, the time gain introduced by the use of PCA or Lloyd & PCA will be multiplicative with the time gain introduced by such improved versions of MDAV, resulting in algorithms that perform still better. This will be part of the future work to be done within this project.

Furthermore, this algorithms cause memory savings, because the reduced-dimension version of the dataset will need less memory to be stored. Therefore, the dataset will be stored in faster memories (for example, it could happen that a dataset that could not be entirely stored in the RAM, after being projected is small enough to be stored there), which should also decrease the execution time.

REFERENCES

[1] L. Sweeney. "Uniqueness of simple demographics in the U.S. population". Technical Report LIDAP-WP4, Pittsburgh, PA, 2000.

[2] A. Oganian and J. Domingo-Ferrer. "On the complexity of optimal microaggregation for statistical disclosure control". Technical report.

[3] S. L. Hansen and S. Mukherjee. "A polynomial algorithm for optimal univariate Microaggregation". IEEE Transactions on Knowledge and Data Engineering, 15(4):1043–1044, July-August 2003.

[4] J. Domingo-Ferrer and V. Torra. "Ordinal, continuous and heterogenerous k-anonymity through microaggregation". 11(2):195–212, 2005.

[5] A. Solanas and A. Martínez-Ballesté. "V-MDAV: a multivariate microaggregation with variable group size". In A. Rizzi and M. Vichi, editors, Proceedings in Computational Statistics COMPSTAT 2006, Heidelberg: Springer's Physica Verlag, pages 917–925, 2006.

[6] John Panaretos. "Aspects of Estimation Procedures at Eurostat with Some Emphasis on Over-Space Harmonisation". Chapter 2.

[7] J. M. Mateo-Sanz, J. Domingo-Ferrer. "A comparative study of microaggregation methods". Data Min Knowl Disc 11(2): 195-212, 1998.

[8] International Data Corporation (IDC)

[9] S. P. Lloyd. "Least squares quantization in PCM," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 129–137, Mar. 1982.

[10] Jolliffe, I.T. "Principal Component Analysis", Springer.

[11] D. Rebollo-Monedero and J. Forné and M. Soriano and J. Puiggalí. "k-Anonymous Micro¬aggregation with Preservation of Statistical Dependence". *(Elsevier) Inform. Sci.*, submitted.